

MASTERS 2013



The premier technical training conference for embedded control engineers

17104 MC2

Running BLDC Motors with the PIC16F1783 and the MCLV-2 Demo Board

Class Agenda

- **Working with the PSMC-Module**
- **Configuring the Analog-Modules**
 - Lab 1- Get the Motor running
- **How to measure the Motor during runtime**
 - Lab 2 – Measuring the Motor
 - Lab 3 – Controlling the Motor
- **Summary**

Class Objectives

When you walk out of this class you will....

- **Know the PSMC Module and use it for spinning BLDC Motors**
- **Know how to work with the analog features of PIC16F1783**
- **Measure a motor during runtime**
- **Get any BLDC-Motor spinning from the scratch**

MASTERS 2013



The premier technical training conference for embedded control engineers

Are you ready to Spin ?



Working with the PSMC Module

Class Agenda

- **Working with the PSMC-Module**
 - Lab 1- Get the Motor running
- Configuring the Analog-Modules
- How to measure the Motor during runtime
 - Lab 2 – Measuring the Motor
 - Lab 3 – Controlling the Motor
- Summary

PSMC Features

● PSMC Output Modes

● Single PWM

- Up to 6 steerable outputs
- Useful for micro-stepping

● Complementary PWM

- Up to 3 pairs of steerable outputs
- Useful for synchronous power supply

● Push-pull PWM

- Multiple pin configurations
- Useful for full and half-bridge configuration

● Pulse Skipping PWM

- With and without complementary output
- Useful for resonant converter

● 3-phase PWM

- Commutates between the 6 different drive sequences

● Fixed Duty Cycle PWM

- Fractional Adjust for better average PWM resolution

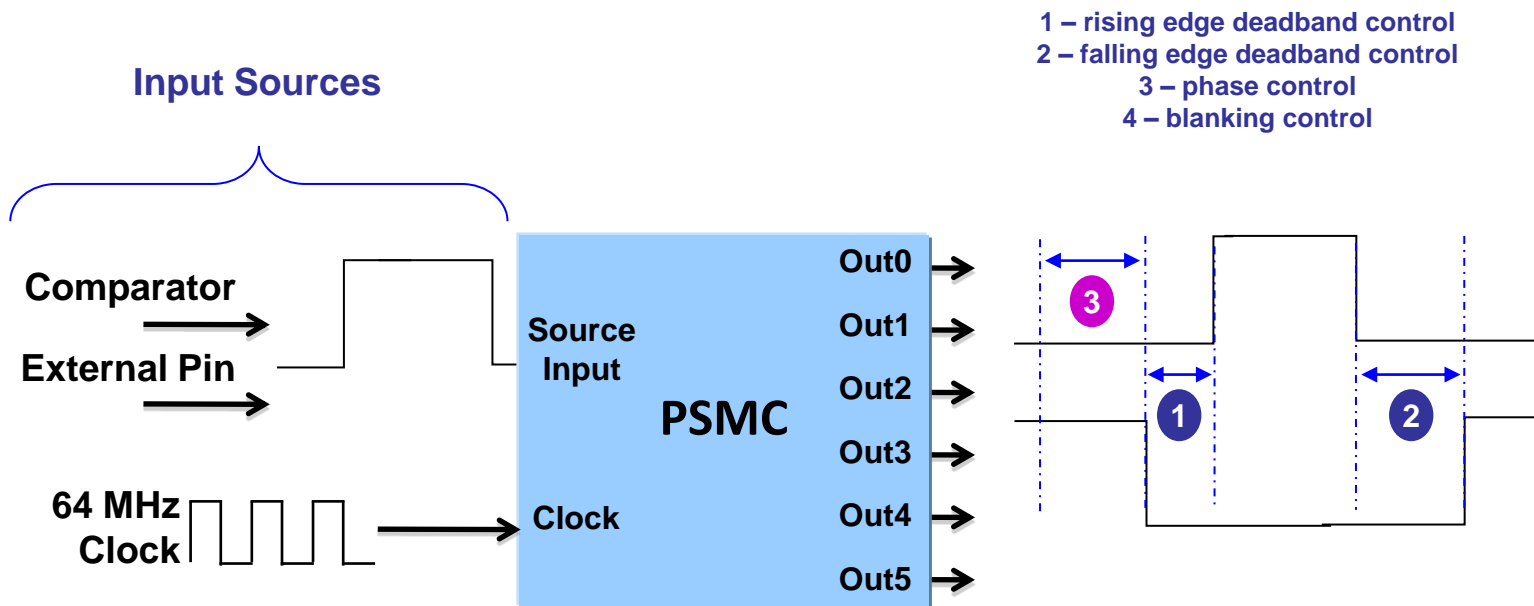
● Brushed DC motor mode (ECCP Compatible)

- Forward and reverse supported



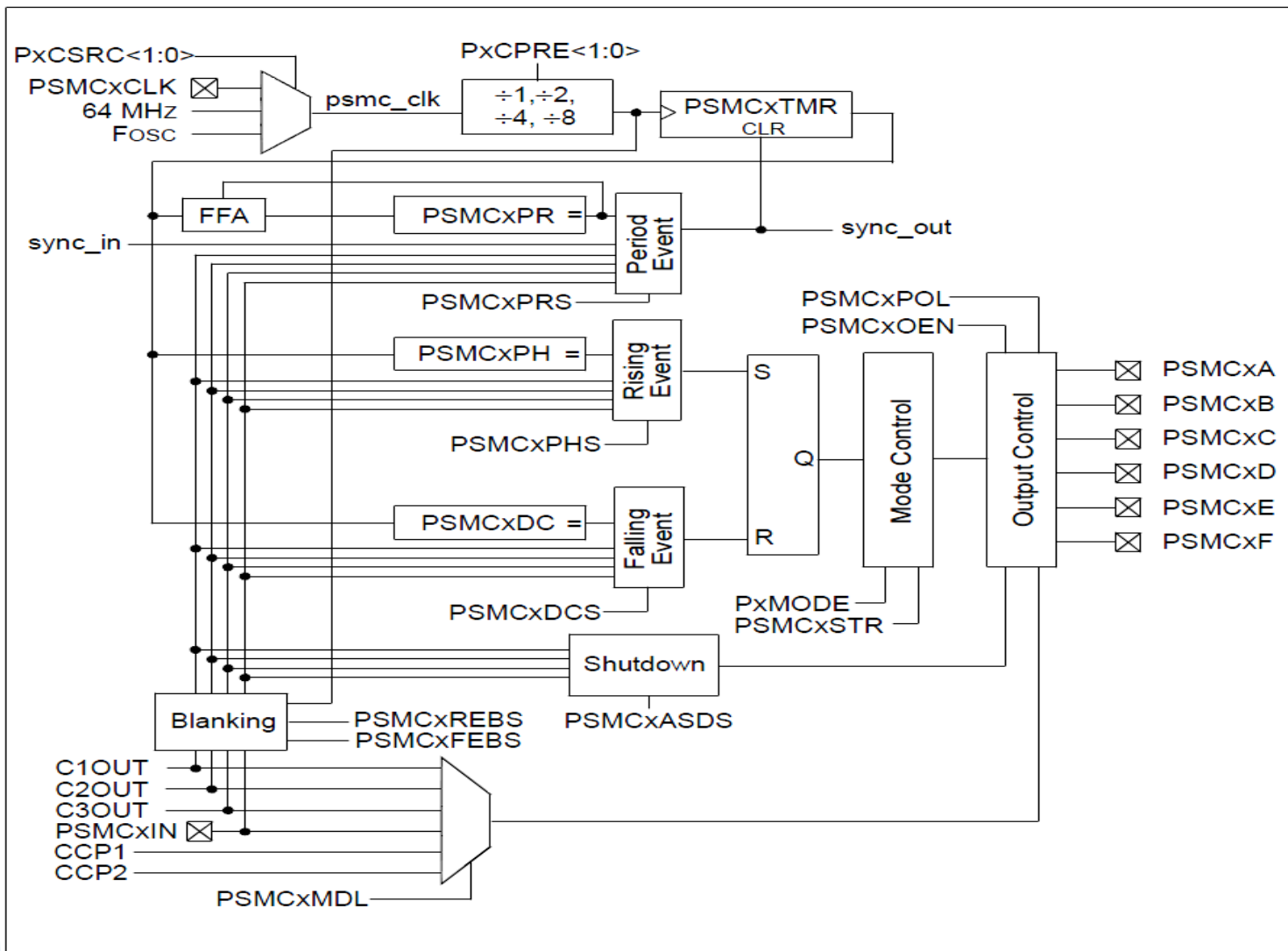


PSMC Introduction





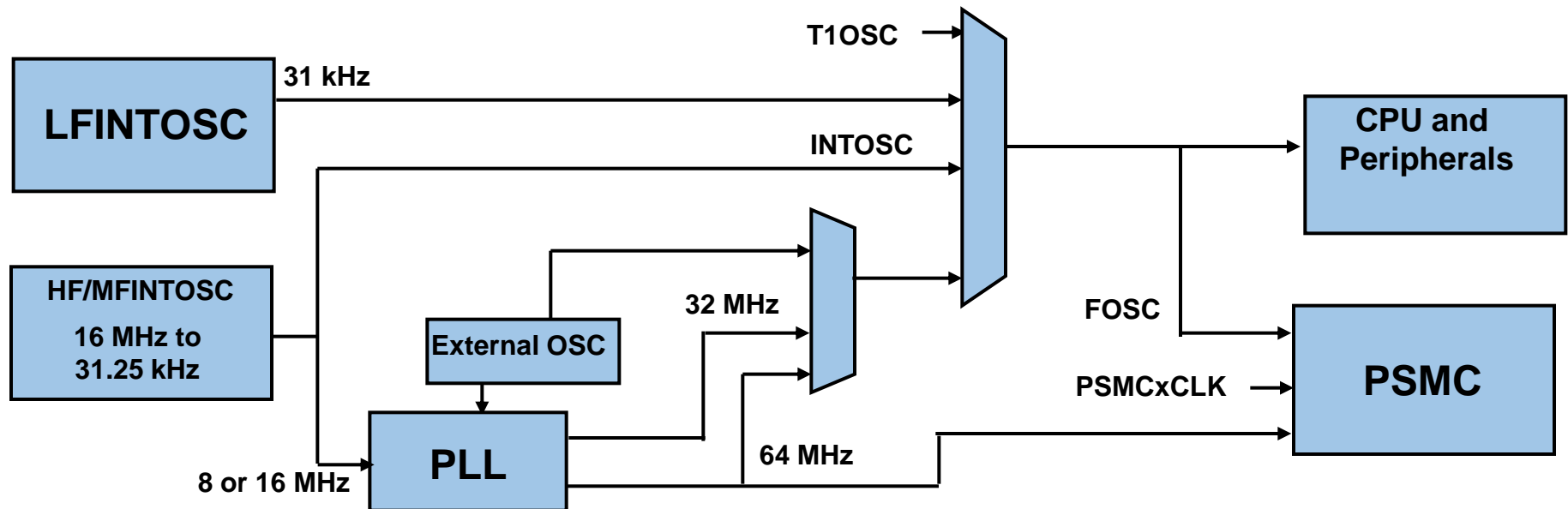
PSMC – Block Diagram





PSMC Clock

- 64 MHz clock input
- Separate clock domain
 - The module is run off of the “psmc_clk” signal
 - This signal is INDEPENDENT of the CPU clock speed ($F_{osc}/4$)
 - Saves power by allowing the peripheral to run at a separate frequency than the CPU

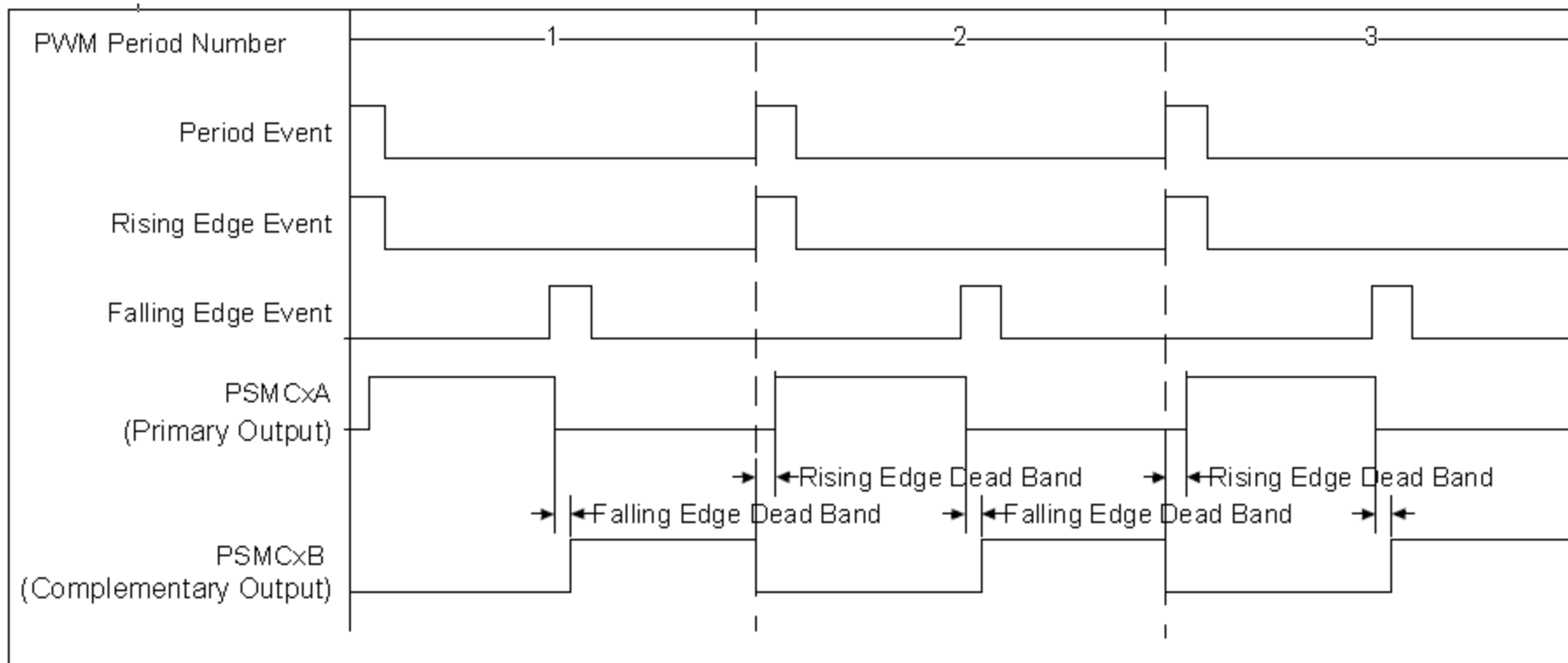




PSMC

Single-Phase Mode:

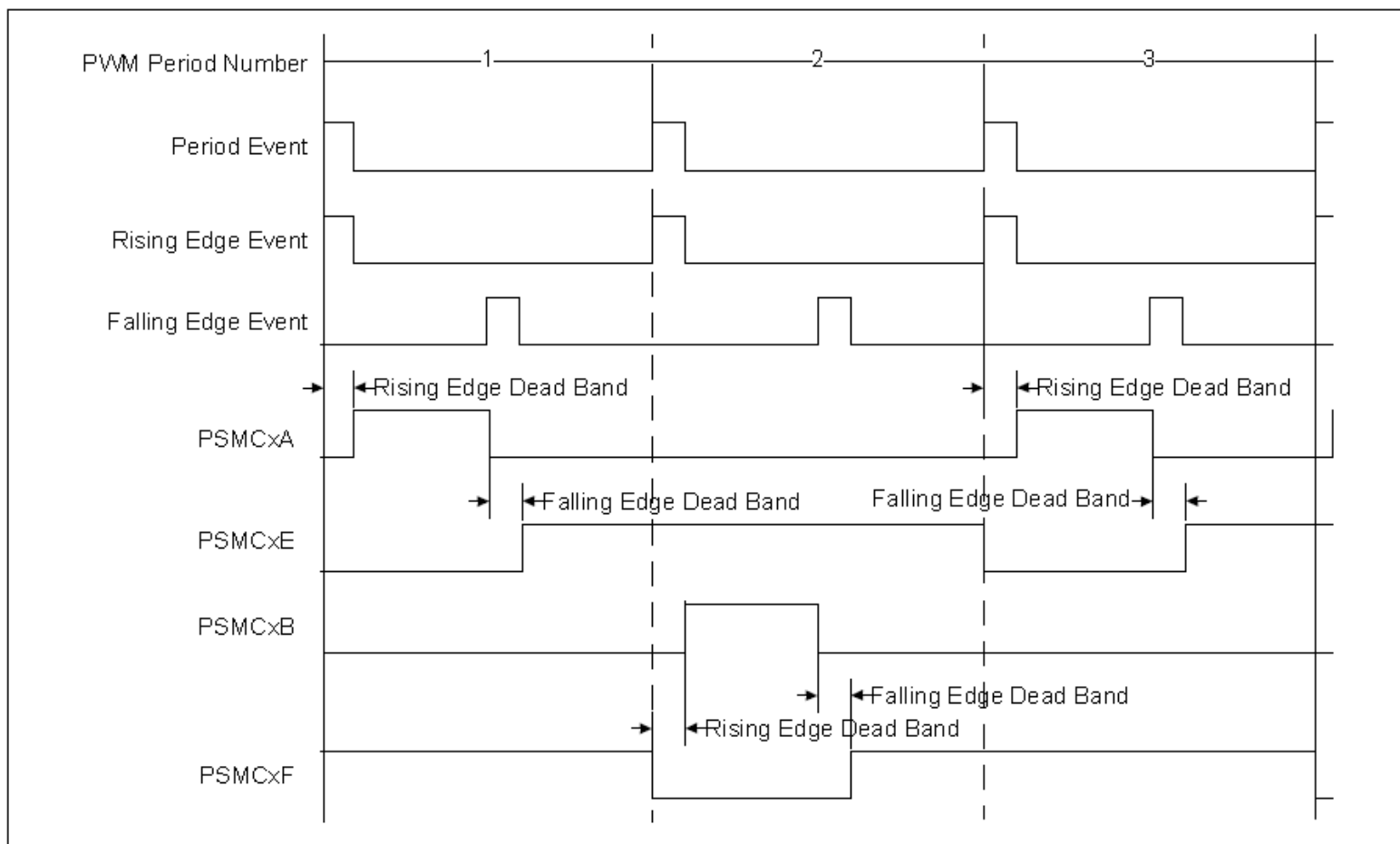
Simple or Complementary Single-Phase Waveform





PSMC

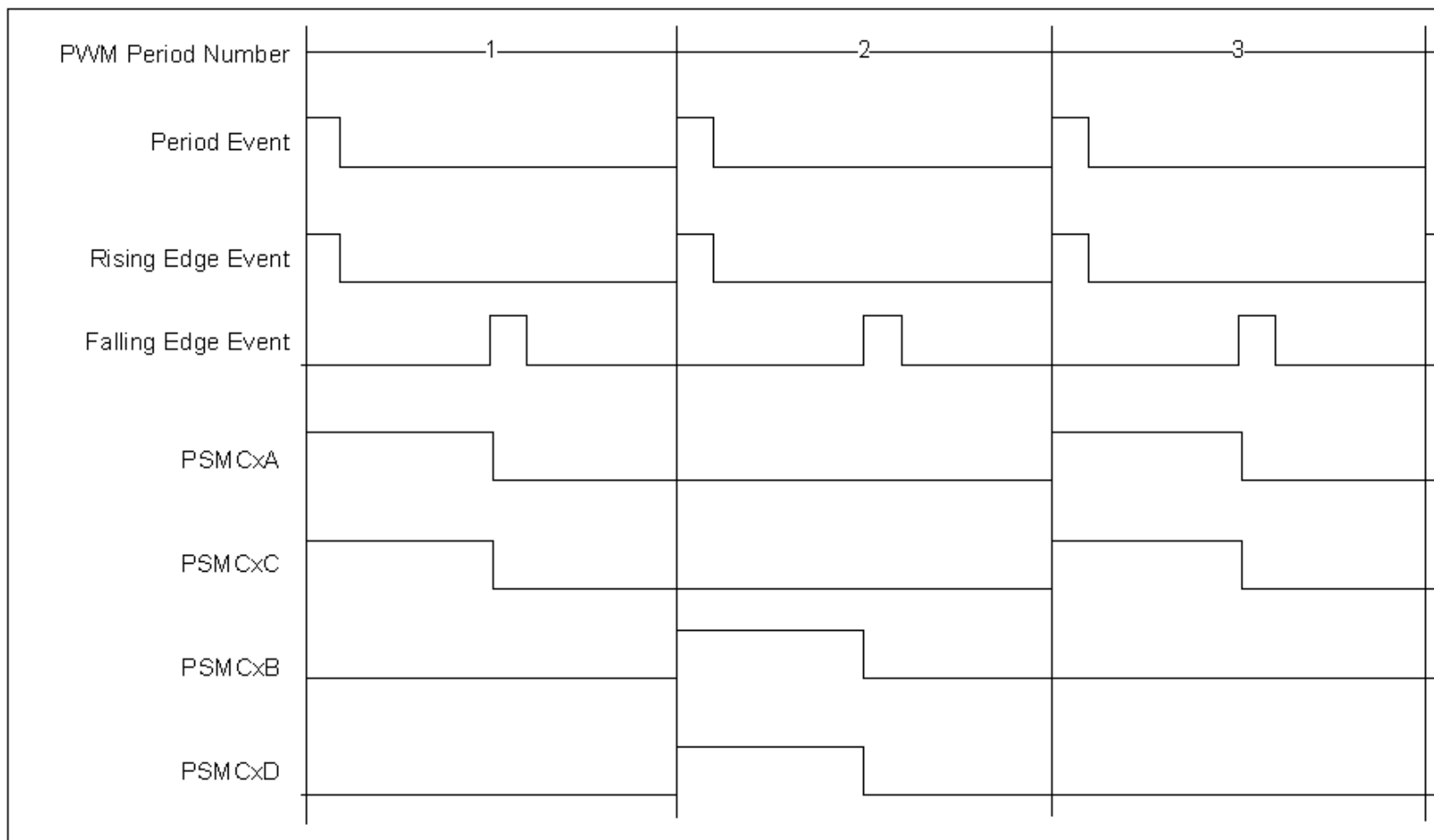
Push-Pull PWM Mode





PSMC

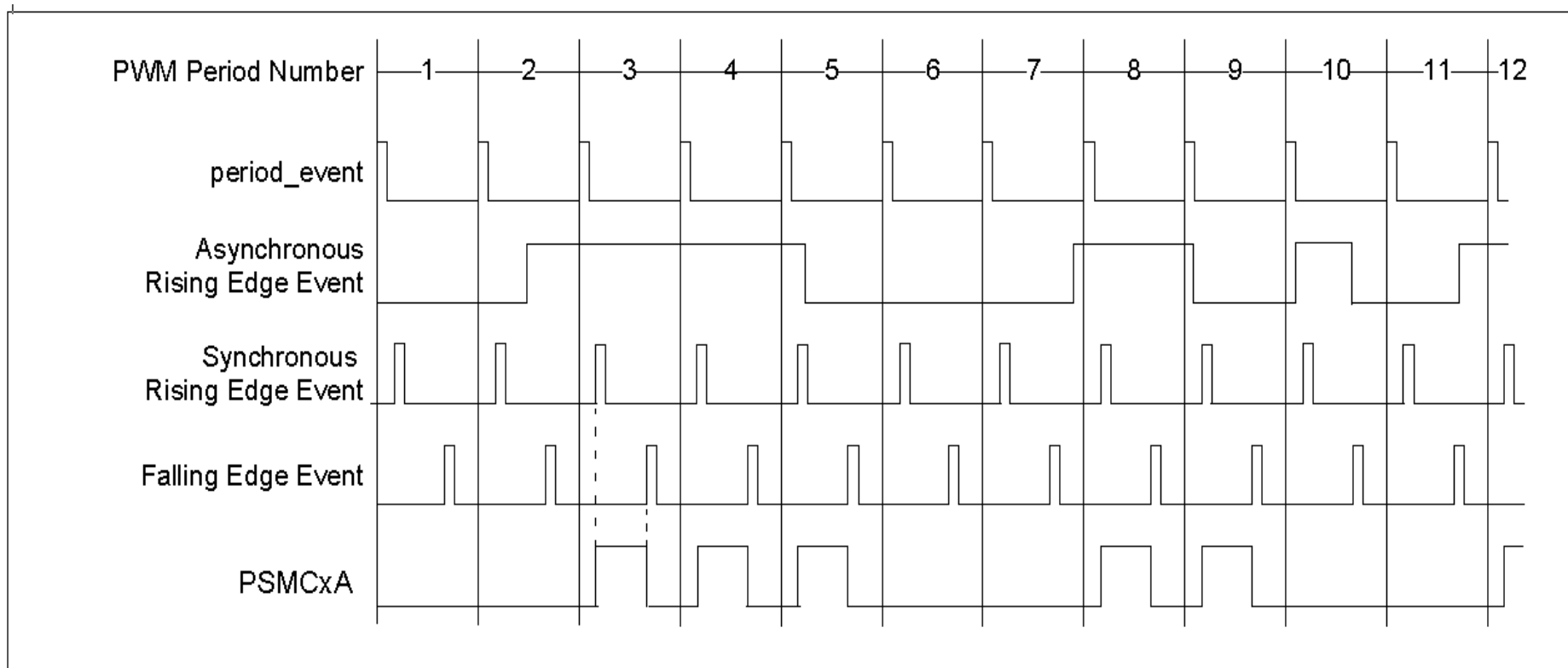
Push-Pull PWM with 4 Full-Bridge Outputs Mode





PSMC

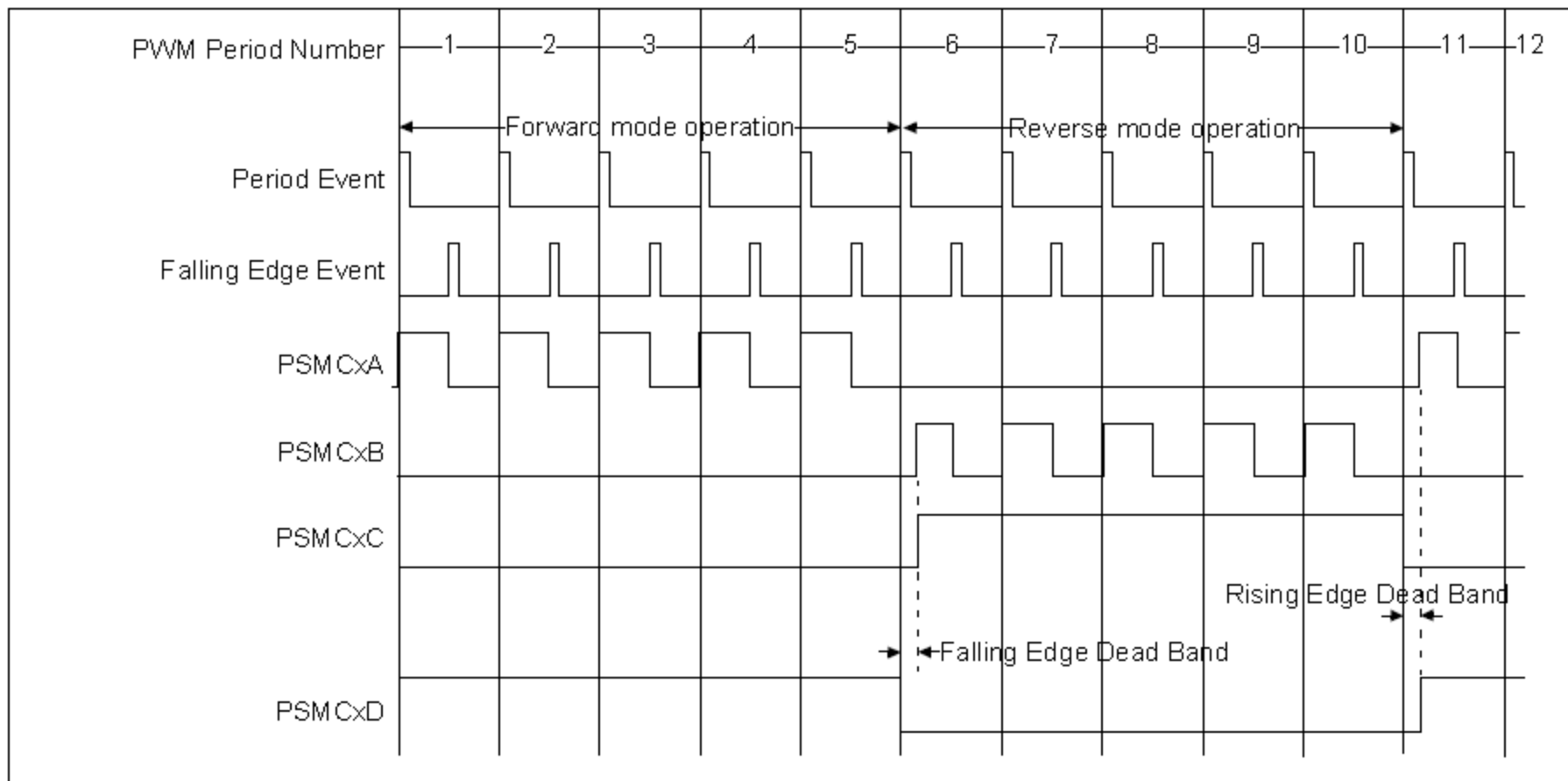
Pulse-Skipping PWM with Complementary Outputs Mode





PSMC

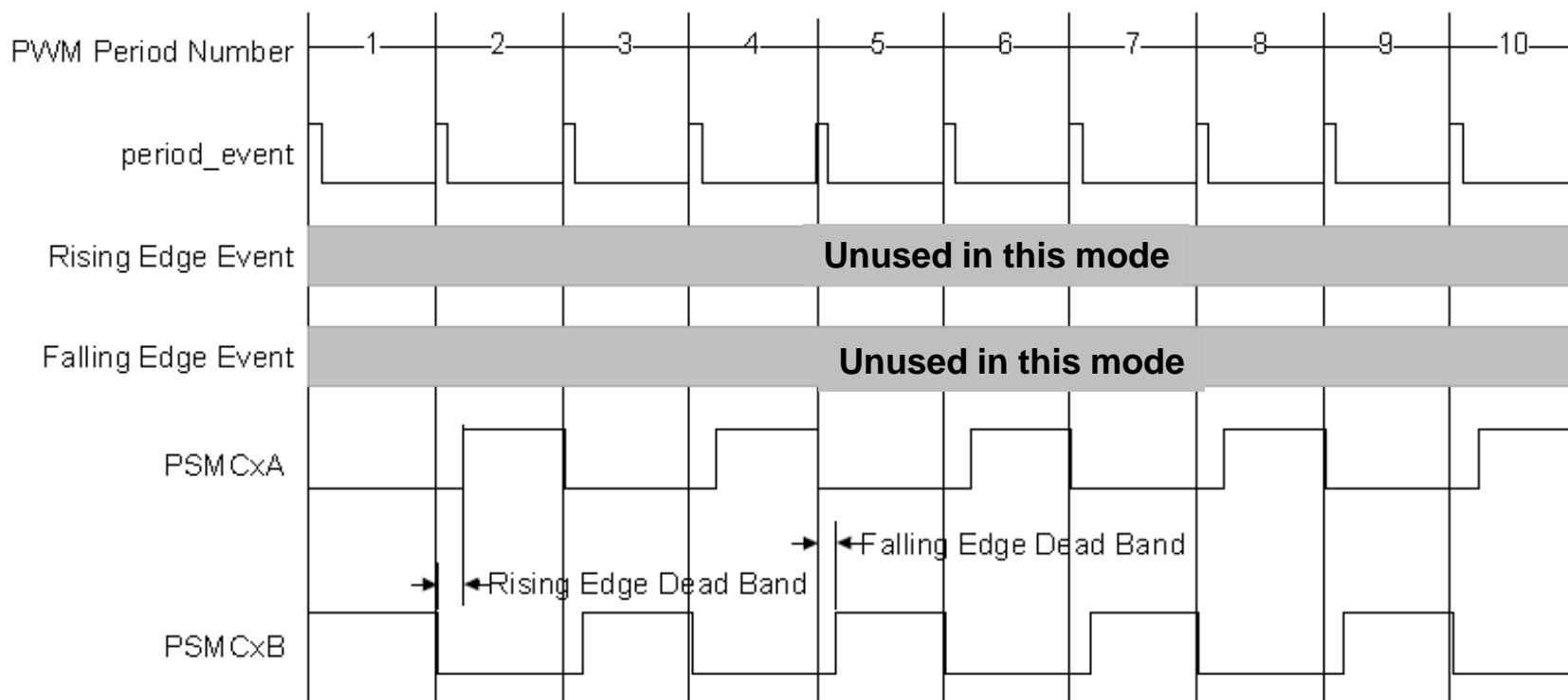
ECCP Compatible Full Bridge PWM Mode





PSMC

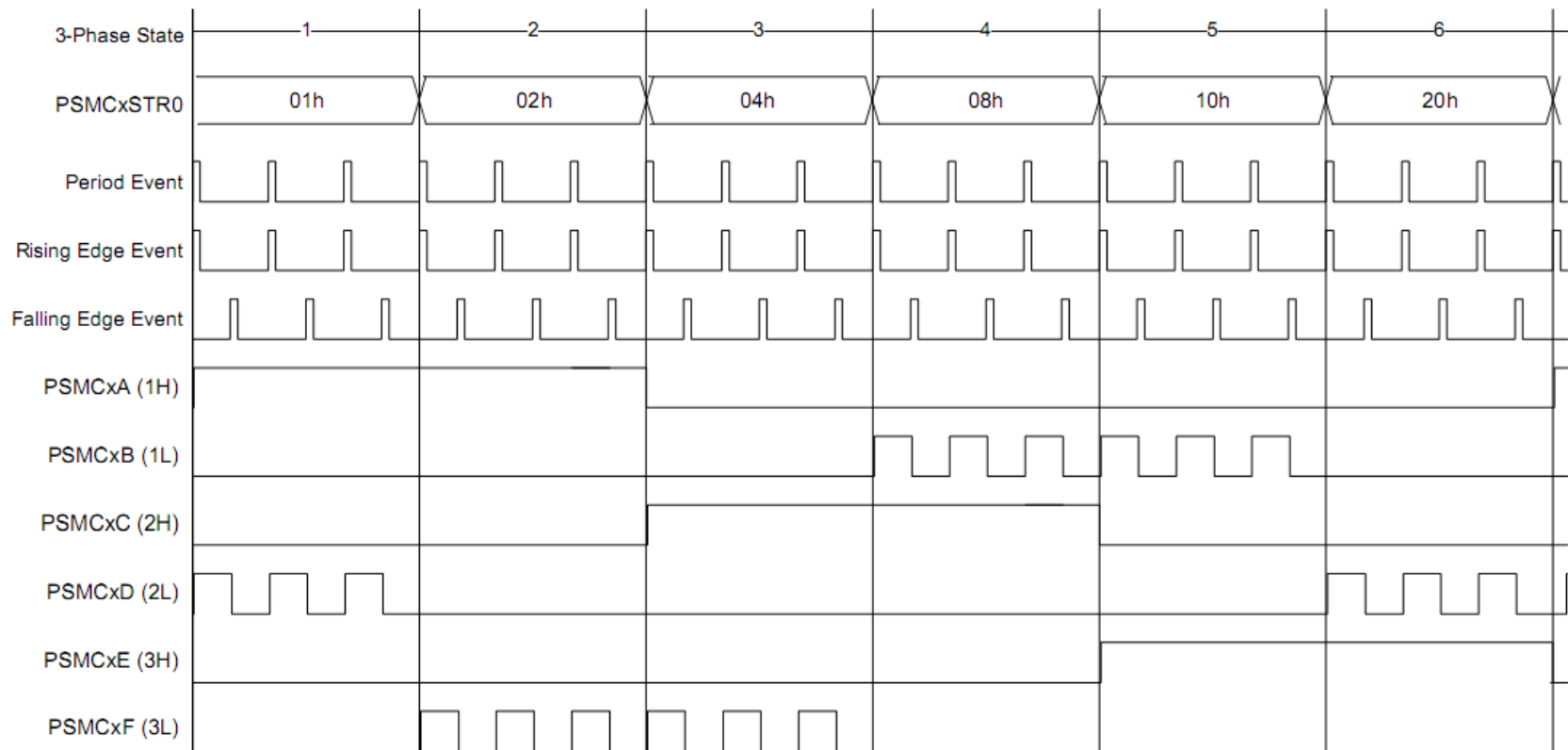
Variable Frequency Fixed Duty Cycle PWM with Complementary Outputs Mode





PSMC

3 Phase PWM Mode Waveform





PSMC Registers

PSMCxCON: MODULE CONTROL

PSMCxEN	PSMCxLD	PxDBFE	PxDBRE	PxMODE<3:0>
---------	---------	--------	--------	-------------

PSMCxEN

- Module enable bit

PxDBFE/PxDBRE

- Falling/Rising Edge Dead-Band enable bit

PxMODE

- Operating Mode bits
- 3 phase
- Pulse-Skipping
- Push-Pull, etc.

PSMC Registers

PSMCxCLK: CLOCK CONTROL

-	-	PxCPRE<1:0>	-	-	PxCSRC<1:0>
---	---	-------------	---	---	-------------

PxCPRE

- Clock Pre-scaler Selection Bits
- /8, /4, /2, /1 Selectable

PxCSRC

- Clock Source Selection
- PSMCxCLK pin
- 64 MHz dedicated clock
- Fosc

PSMC Registers

PSMCxPHS: PHASE (RISING EDGE) SOURCE

PxPHSIN	-	-	-	PxPHSC3	PxPHSC2	PxPHSC1	PxPHST
---------	---	---	---	---------	---------	---------	--------

PxPHSIN

- Rising Edge on PxPHSIN pin

PxPHSCy

- Rising Edge Occurs on Comparator y output

PxPHST

- Rising Edge Occurs on Timer base match
- PSMCxTMR=PSMCxPH

PSMC Registers

PSMCxDCS: DUTY CYCLE (FALLING EDGE) SOURCE

PxDCSIN	-	-	-	PxDCSC3	PxDCSC2	PxDCSC1	PxDCST
---------	---	---	---	---------	---------	---------	--------

PxDCSIN

- Falling Edge on PxDCSIN Pin

PxDCSCy

- Falling Edge Occurs on Comparator y output

PxDCST

- Falling Edge Occurs on Timer base match
- $PSMCxTMR = PSMCxDC$

PSMC Registers

PSMCxPRS: PERIOD EVENT SOURCE

PxPRSIN	-	-	-	PxPRSC3	PxPRSC2	PxPRSC1	PxPRST
---------	---	---	---	---------	---------	---------	--------

PxPRSIN

- Period Event on PxPHSIN pin

PxPRSCy

- Period Event Occurs on Comparator y output

PxPRST

- Period Event Occurs on Timer base match
- $PSMCxTMR = PSMCxPR$

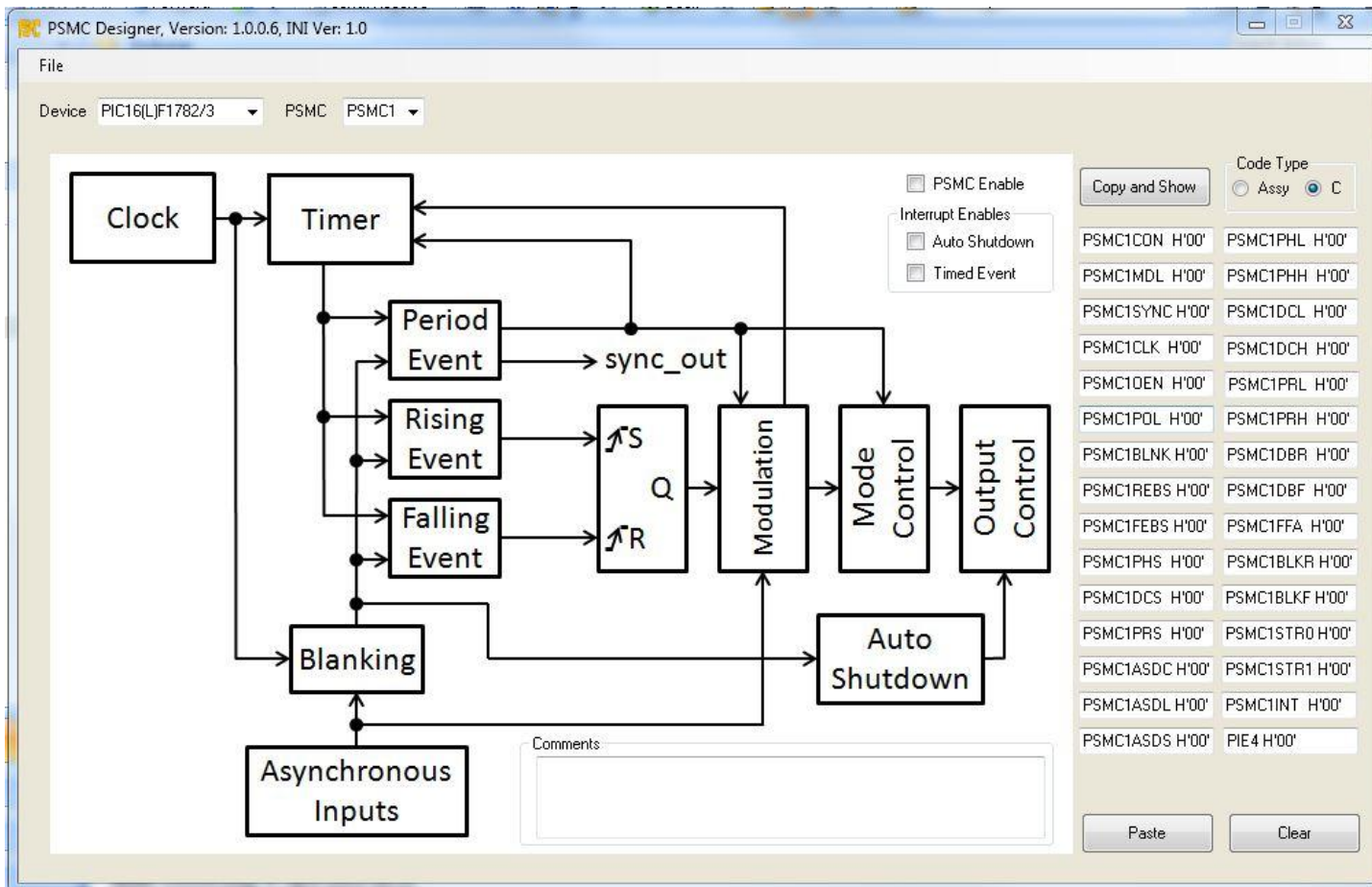


MICROCHIP

MASTERS 2013

Working with the PSMC-Designer

Yes, there is a GUI and it looks like:



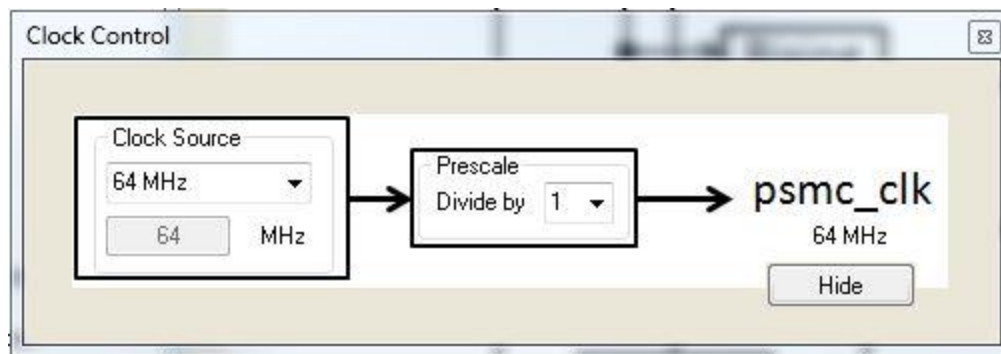
PSMC-Designer

- **Select Device → PIC16F1783**
- **Select PSM-Module → PSMC1**
- **Set PSMC Enable**
- **Adjust CLK, Events, Mode etc.**
- **Press „Copy and Show“**
- **Check the Result**
- **Save the .inc-File**
- **Include the Result in your Project**
- **→ That's it !!!**



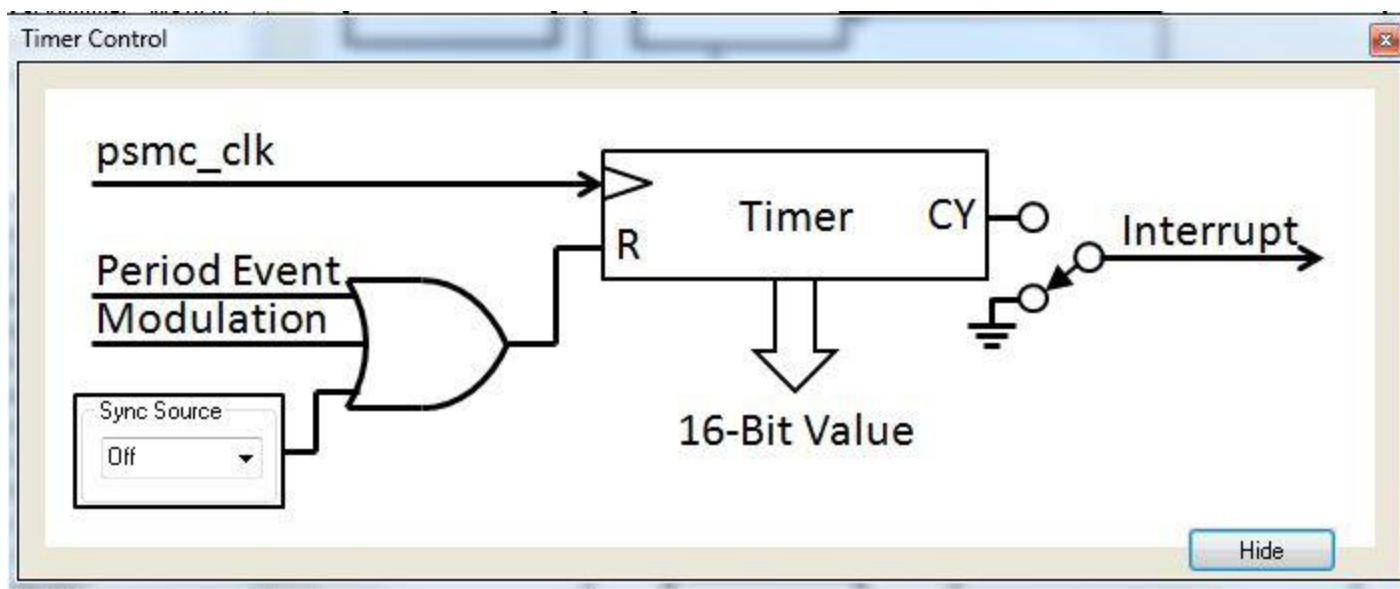
After selecting the MCU and selecting the Module Click on the Clock-Block

(Cursor will change to a pointing Hand on each Block, which can be adjusted)



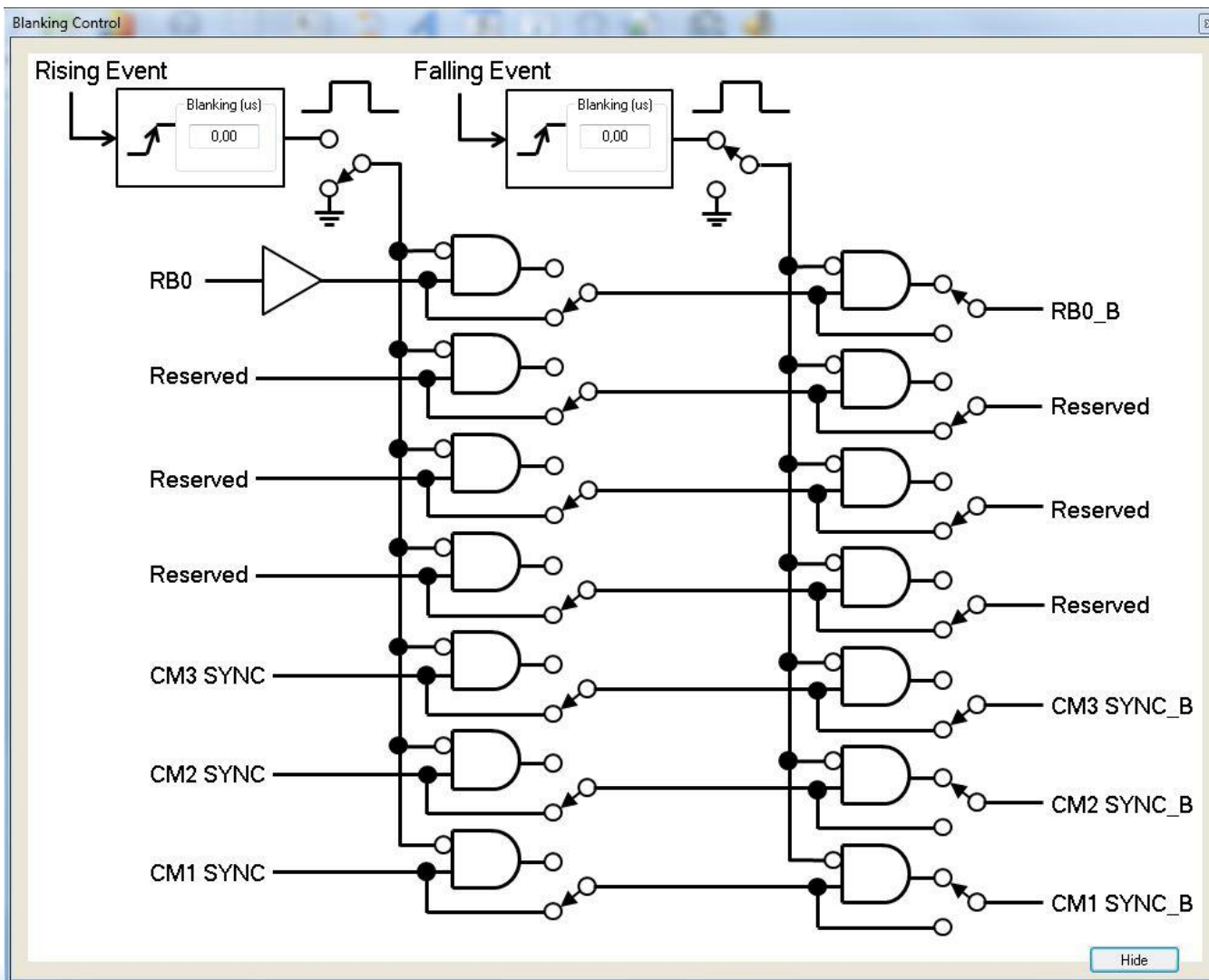


Same for the Timer



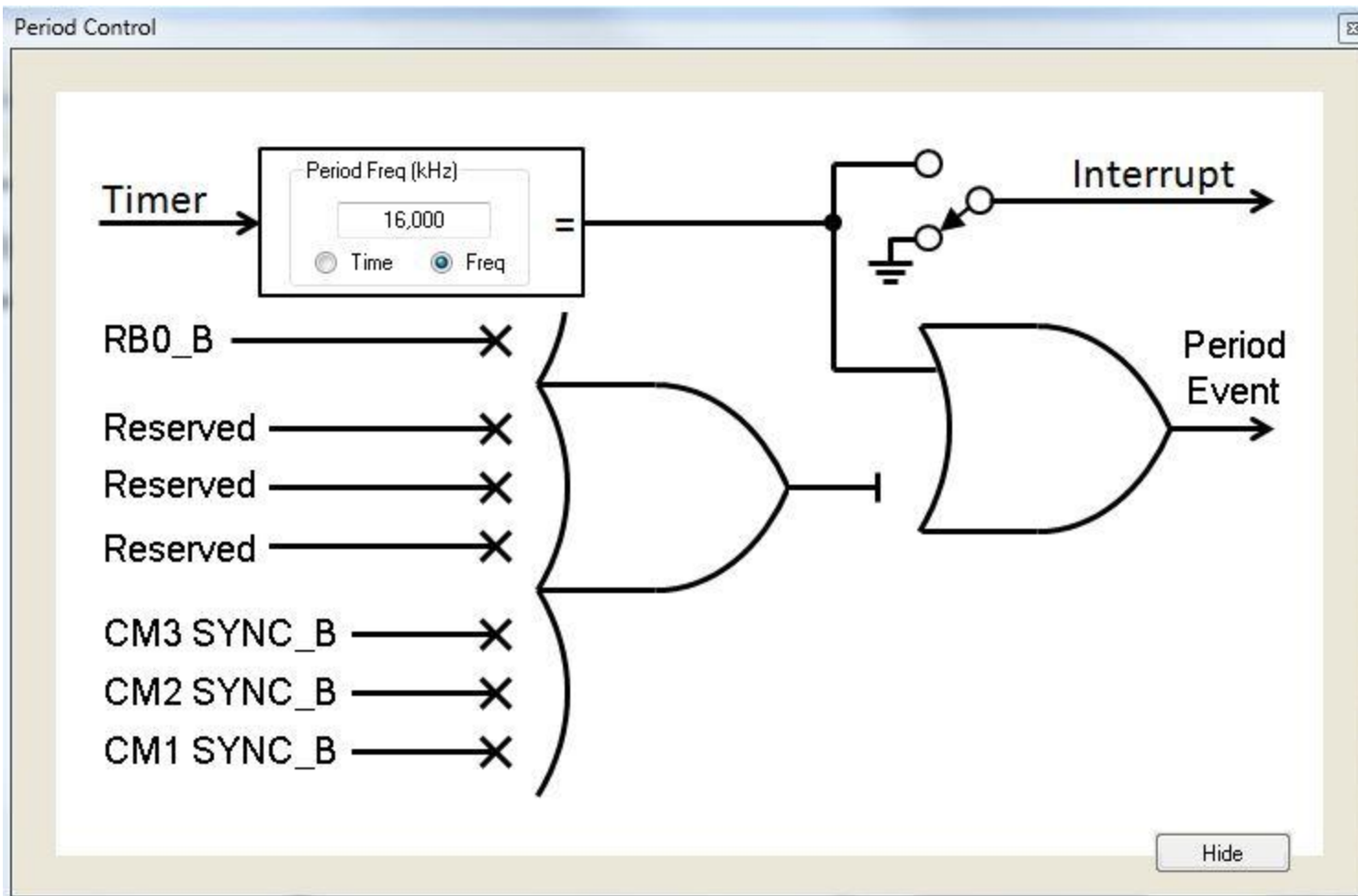


Blanking Adjustments



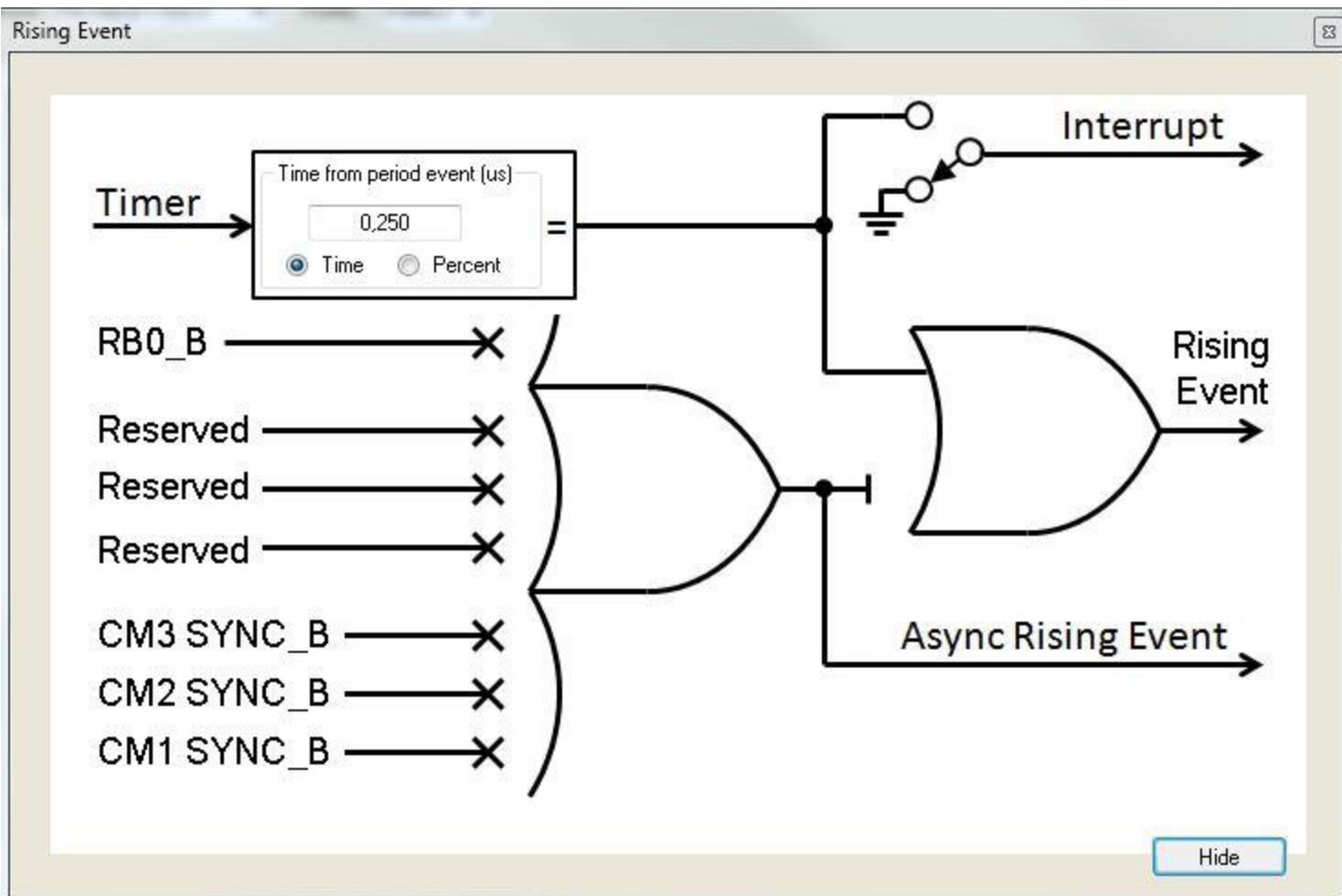


Adjust the Period to 16kHz, no Interrupt



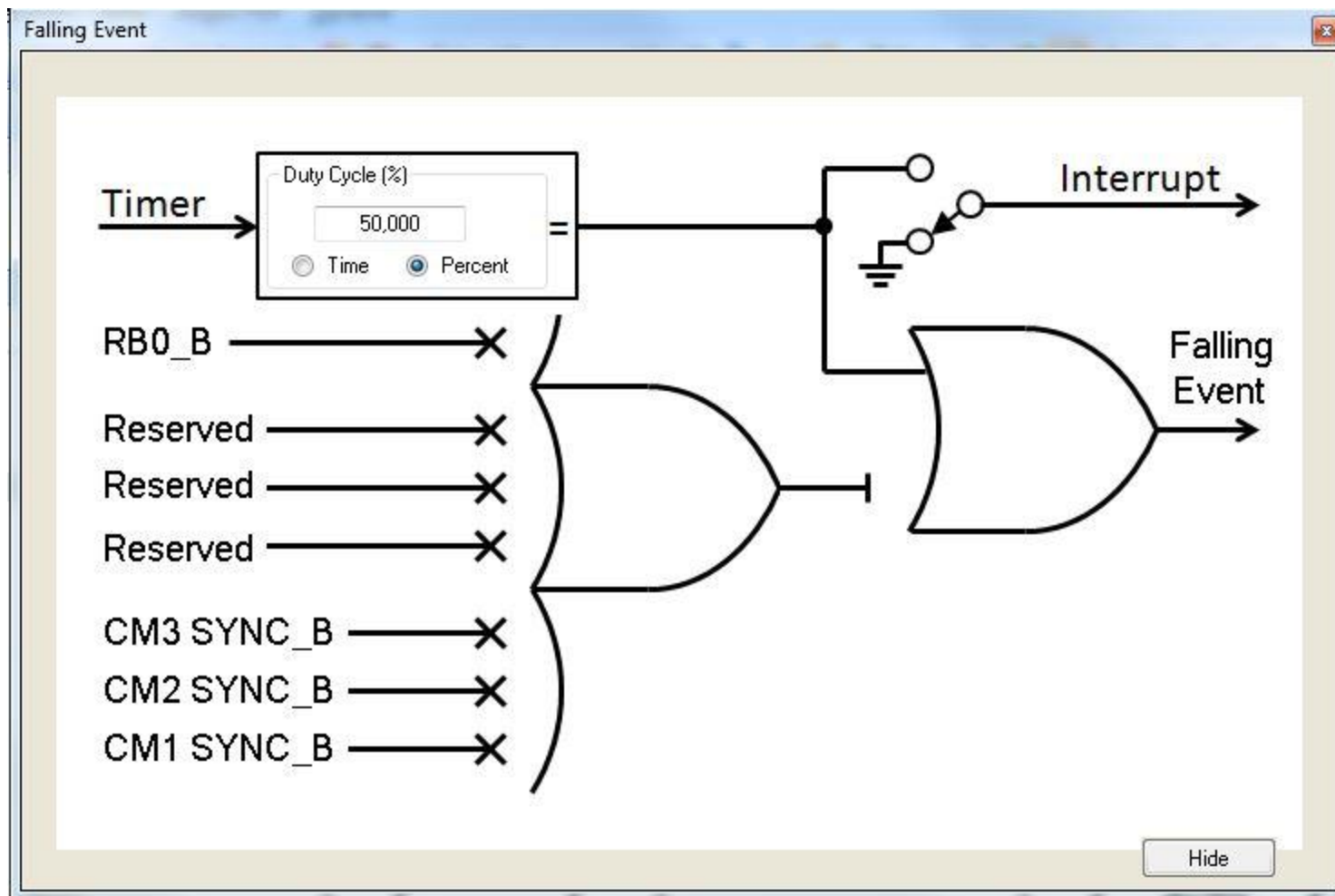


To prevent Shortcuts, adjust a slight Delay for the Rising Event: $0.25\mu\text{S}$

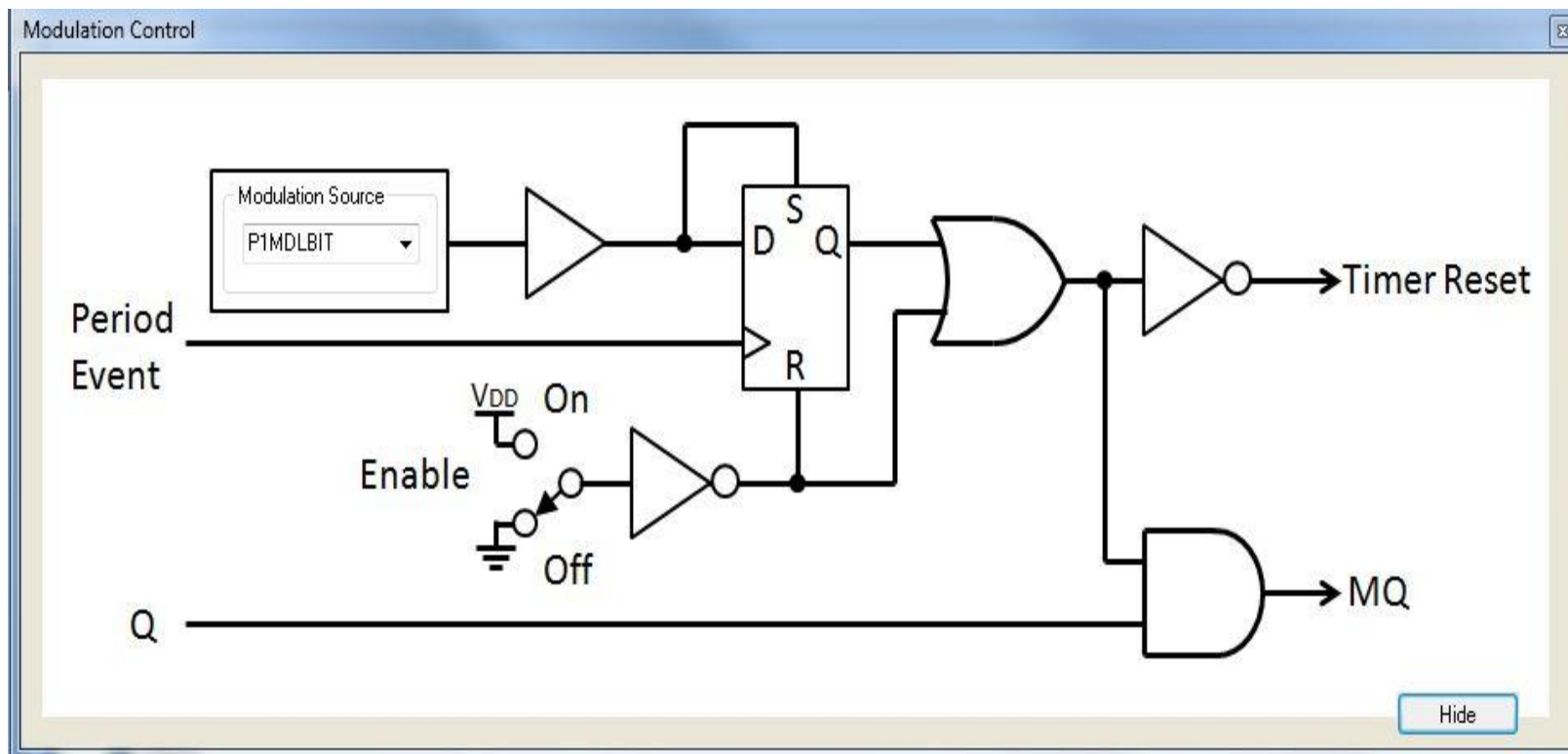




With the Falling Event you can adjust the Duty-Cycle @ 50%

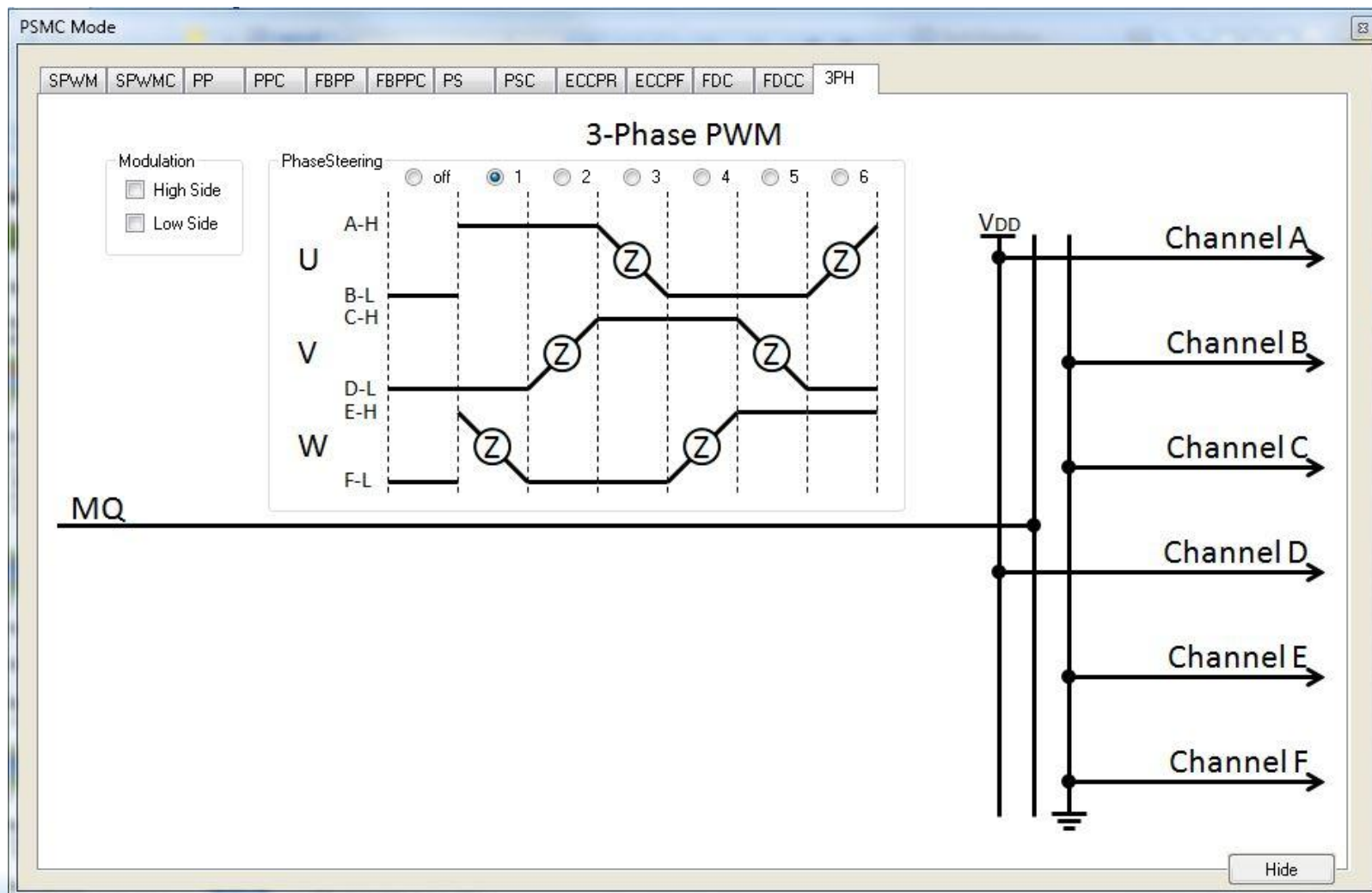


In this case, no modulation is needed



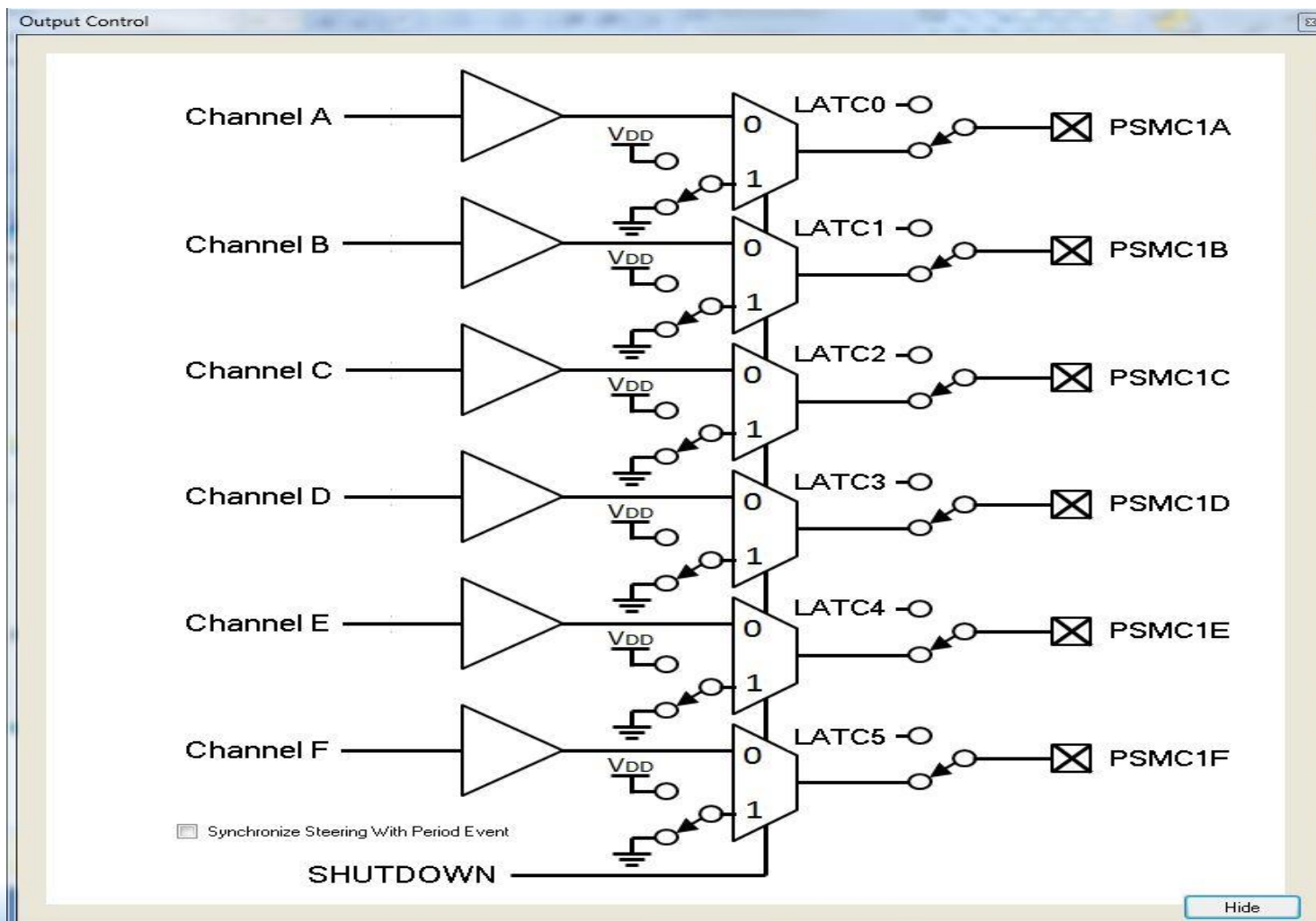


Select 3PH with Starting Condition @ Phase 1



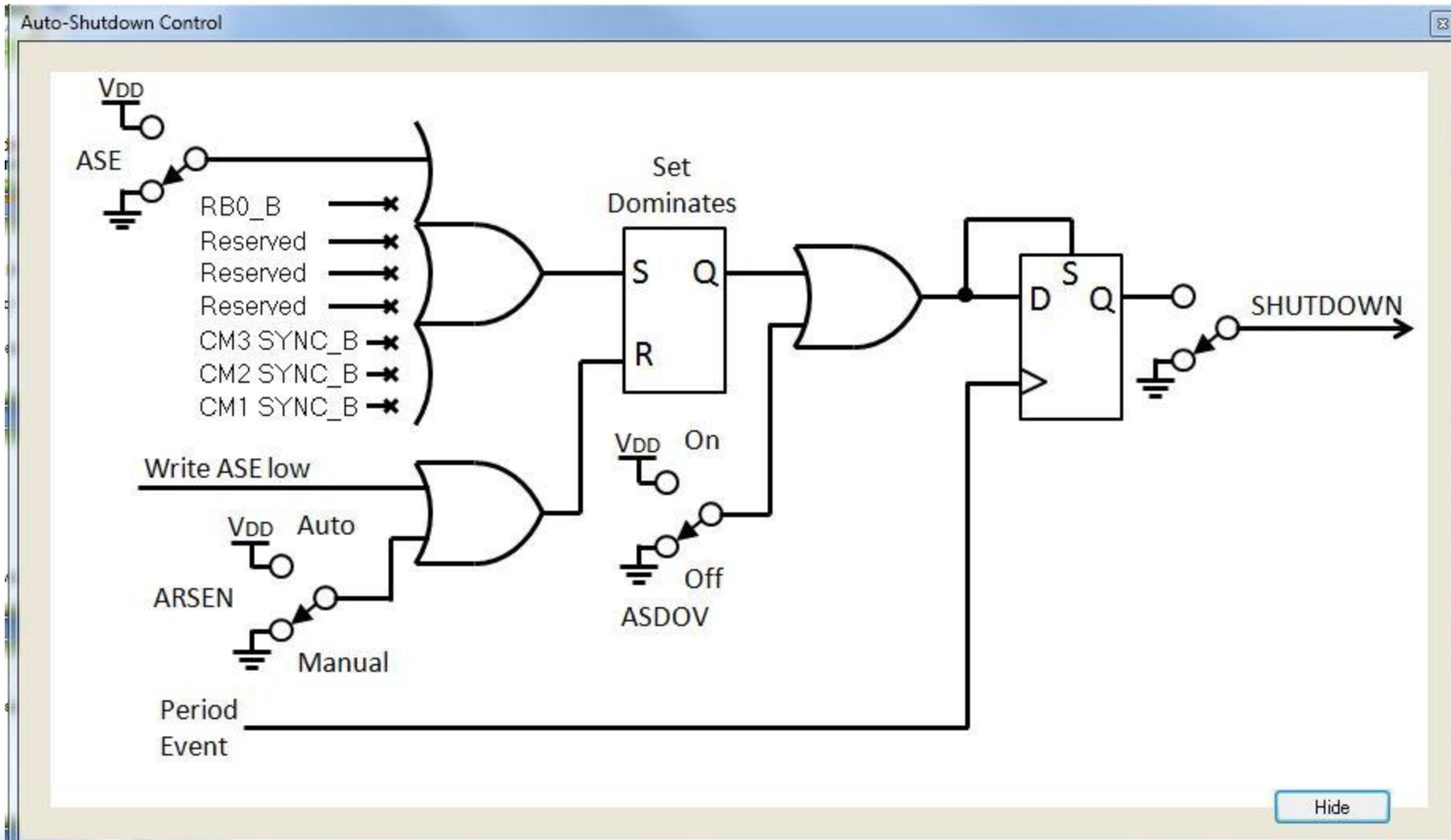


Select the Outputs to be controlled and the Shutdown Condition





For the first Test disable Autoshtutdown

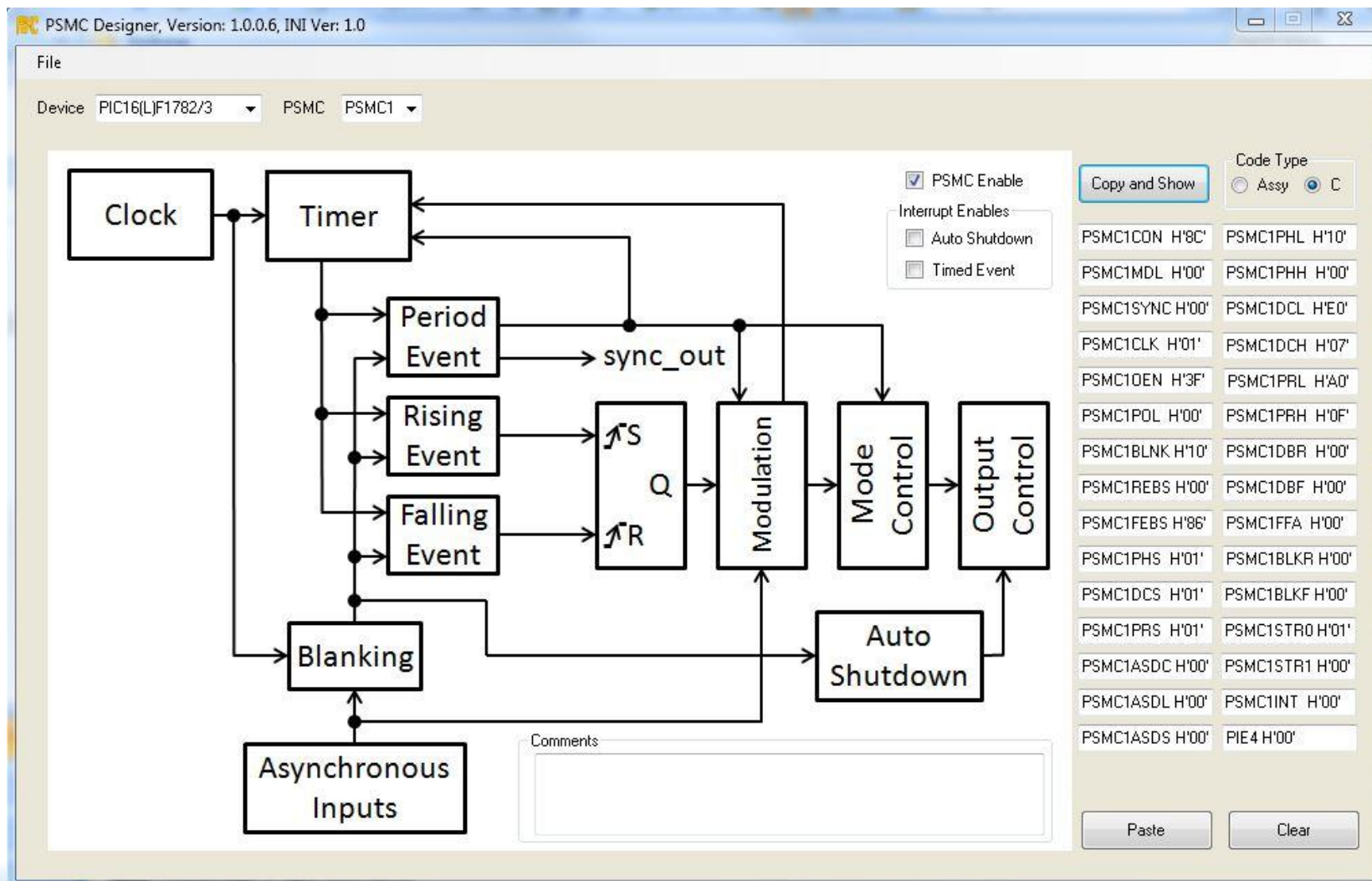




MICROCHIP

MASTERS 2013

In the end, it should look like this:





Lab 1:

Let's spin the Motor

Lab 1 Objectives

- **Give quick look at the demo board**
- **Show which registers need to be set for operation**
- **Show the Motor spinning in an easy way**
- **Think about a better way to control the Motor**

First Result of the Observation

- **What about the Efficiency?**
 - Is the Motor getting hot?
 - How can we avoid this?
- **Controlled Commutation**
 - How to do BEMF-Measurement

Lab 1 Summary

- **PSMC can be adjusted by a GUI**
- **Motor Runs at desired Speed**
- **Bad Efficiency without Control**
- **Large Room for Improvement**

POP Quiz!

- **What is the max speed of the PWM Oscillator?**
 - 32 Khz
 - 8 Mhz
 - 16 Mhz
 - 64 Mhz



Advanced Analog Peripherals

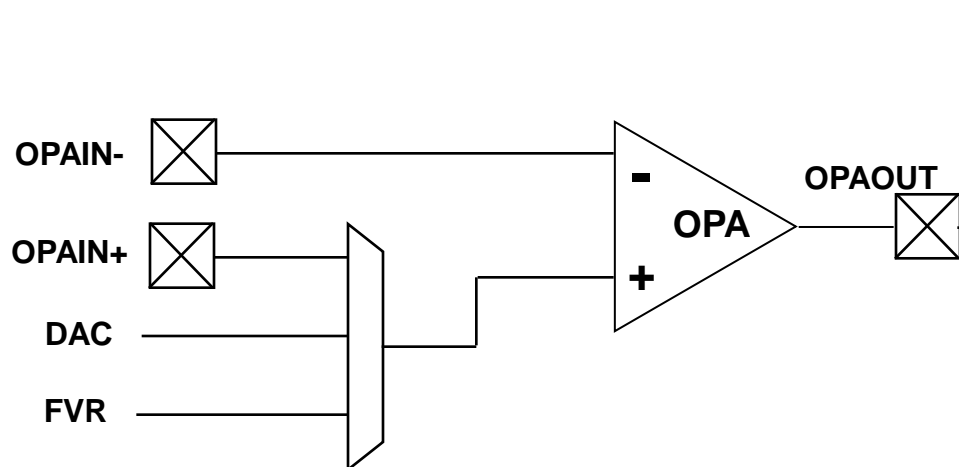




Advanced Analog

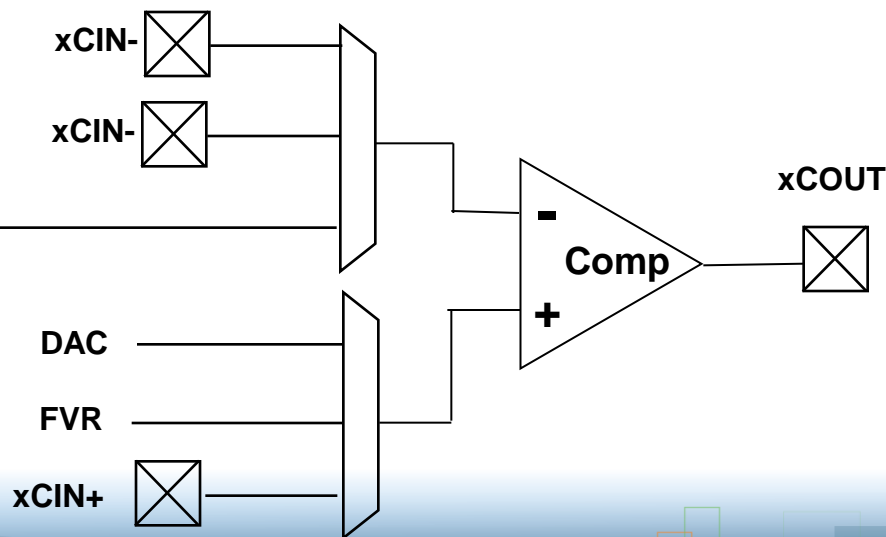
- **Operational Amplifiers**

- **Selectable Gain Bandwidth Product**
 - 100 kHz (typ) Low Power Mode
 - 3 MHz (typ) High Power Mode
- **Input Offset < 5 mV**
- **High current capability**
 - 10 mA (3 MHz GBWP)
- **Common-mode input range**
 - GND to VDD – 1.4V



- **High-Speed Comparators**

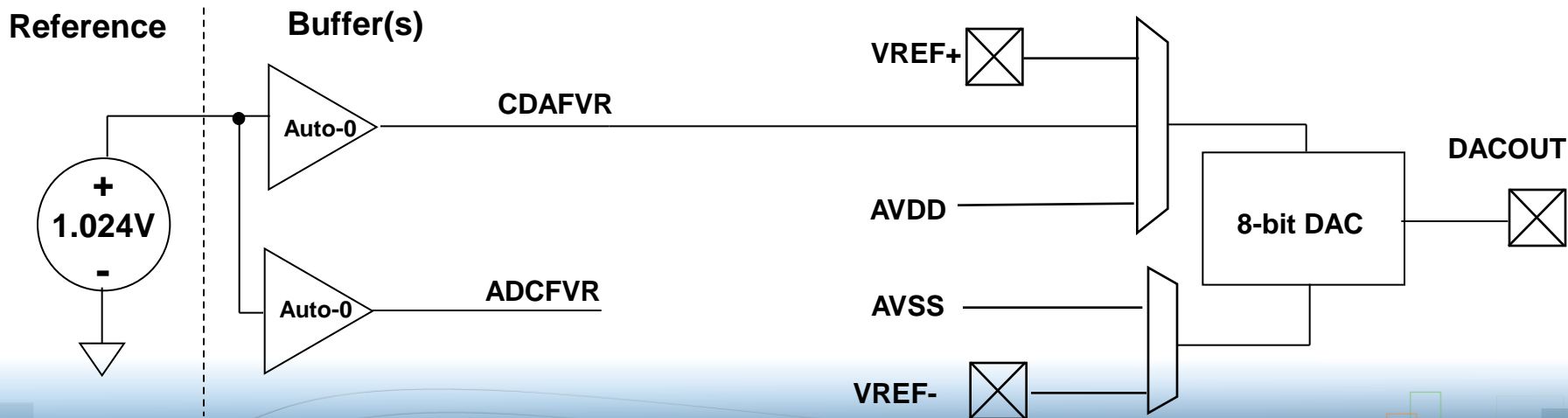
- **Response time**
 - 50 ns (typ) High Power Mode
- **Hysteresis**
 - 25 mV (typ) High Power Mode
- **Input Offset < 10 mV (Max)**
- **Internal connections to OPAMP, DAC and PSMC**





Advanced Analog

- **Improved FVR**
 - Consists of two parts
 - Reference
 - Buffer
 - Using Improved Reference to reduce variation over temp and voltage
 - Using auto-zeroing buffer to improve buffer accuracy
- **8-bit DAC**
 - Resistive DAC
 - No output buffer
 - Output enable
 - Voltage References
 - Positive
 - Negative

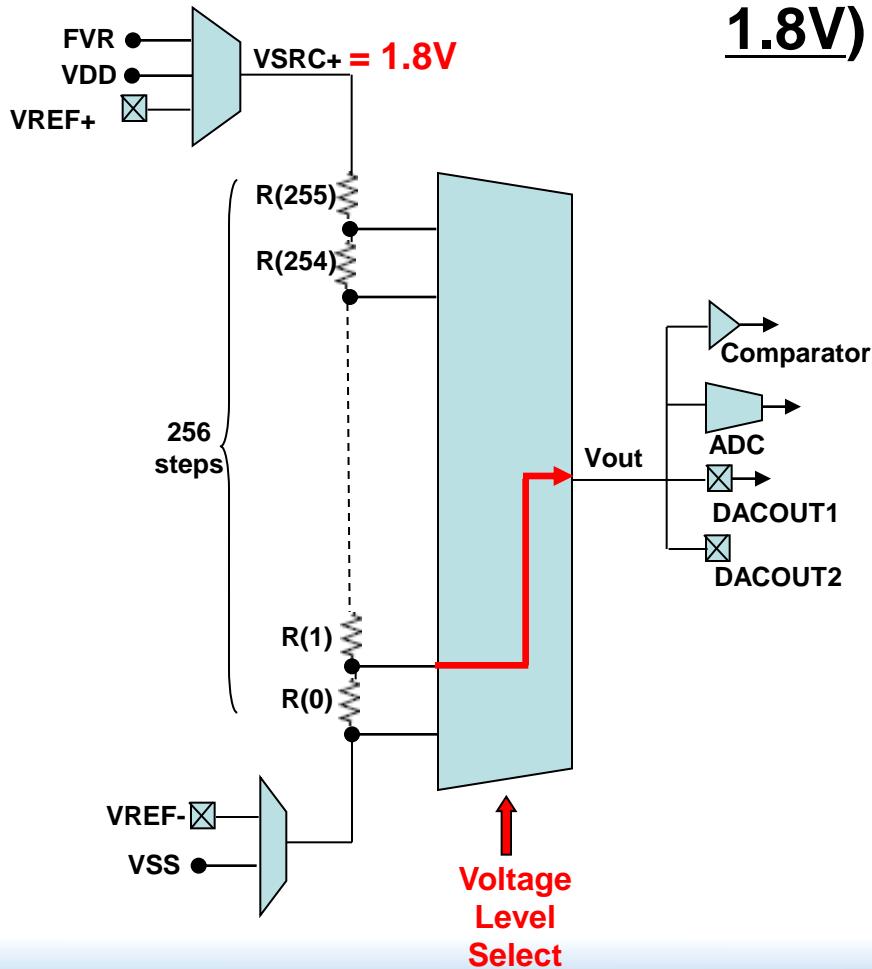




8-bit DAC

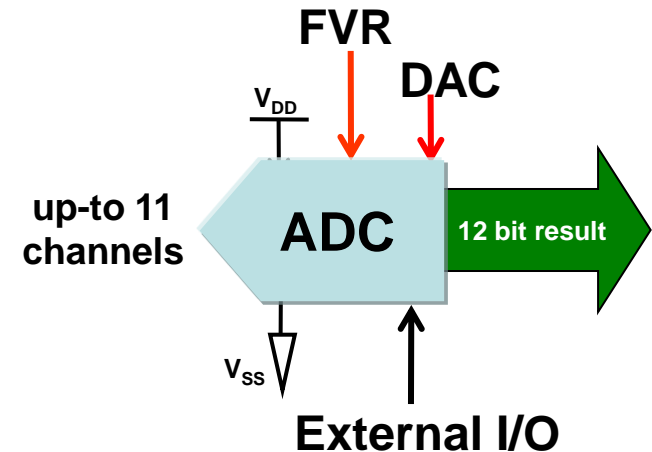
- Full Range

- Vout (Step Resolution @ 1.8V) = 7 mV



12-Bit ADC

- **12-bit ADC**
 - Up-to 11 input channels
 - **5 differential channels**
 - Selectable References
 - **FVR, DAC, or external**
 - 100ksps 12b conversion rate





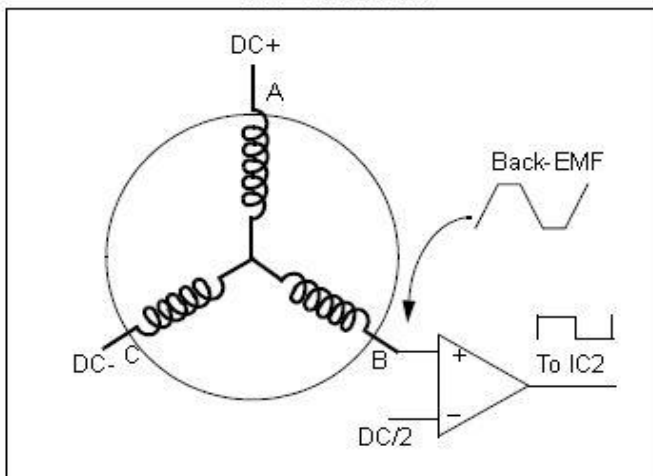
Lab 2:

How to capture BEMF-Signals



BEMF Capturing

Comparison BEMF and Vbus/2



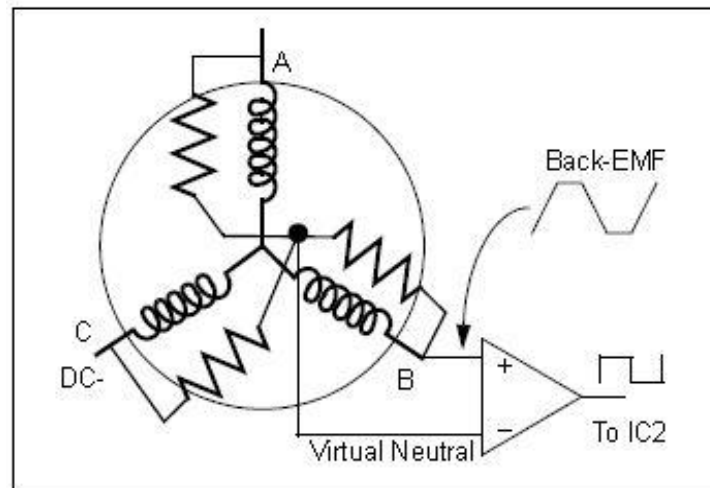
The drawbacks of this method are:

- This method assumes that the motor windings parameters are identical.
- The sensed BEMF signals have positive and negative phase shifts.
- Motor-rated voltage is less than the VDC voltage most of the time; therefore, the zero crossing event not always occurs at VDC/2.

Creating a virtual Neutral-Point and Compare with BEMF

Comparing the BEMF Voltage to the Motor Neutral Point. The zero crossing sensing method described before can be improved by having a variable threshold voltage point used to detect the zero crossing events.

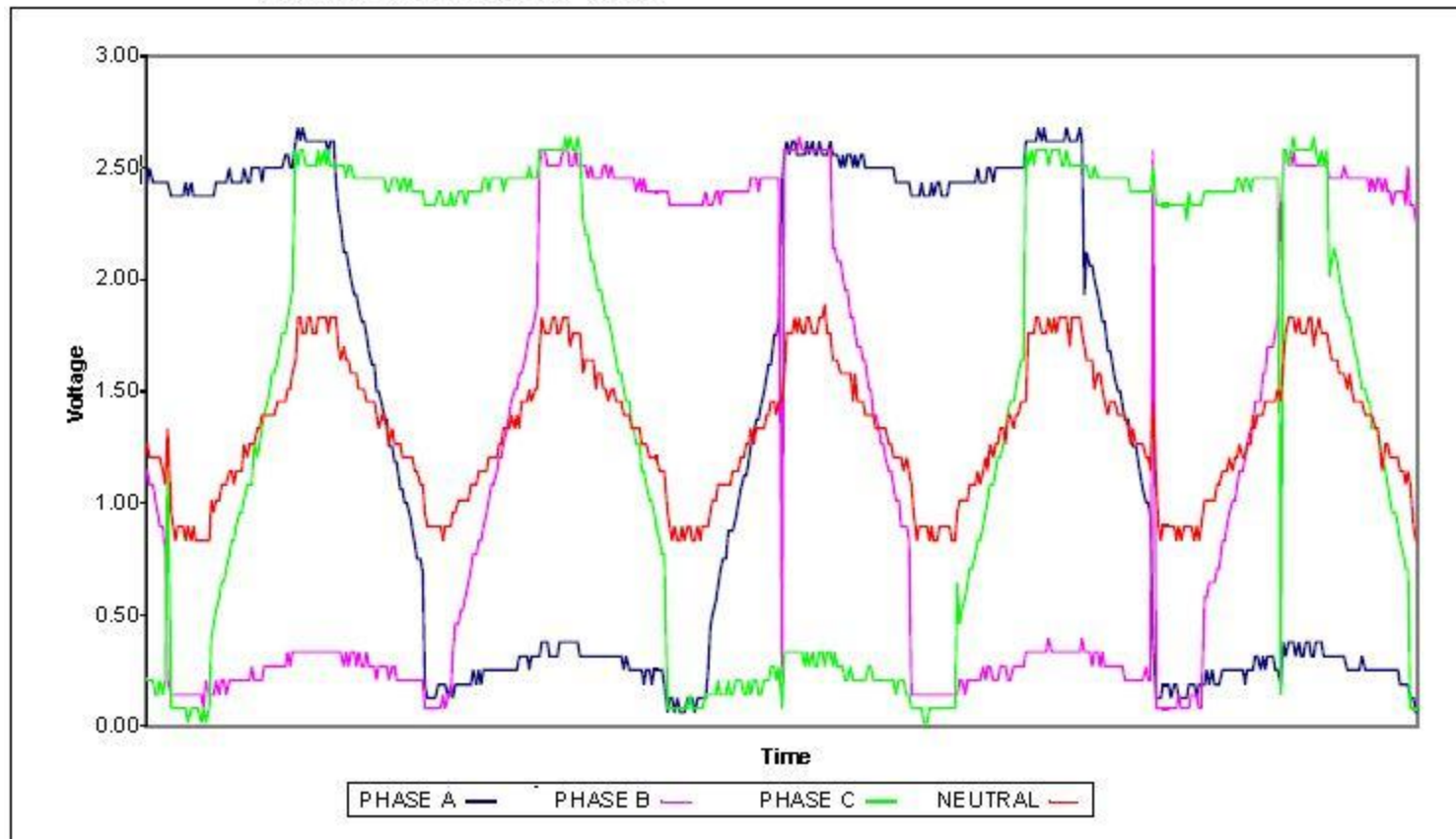
This variable voltage is in fact the motor neutral point. Often, the motor manufacturers do not wire the motor neutral point. However, it can be generated by using a resistor network. Three networks are connected in parallel with the motor windings and connected together to generate a virtual neutral point. This is done on the MCLV-2 Board !





Analyzing BEMF

FIGURE 6: BEMF SIGNALS VERSUS VIRTUAL NEUTRAL POINT WHEN THE PWM DUTY CYCLE IS EQUAL TO 100%





MICROCHIP

MASTERS 2013

Scope Picture

MSO-X 3034A, MY52160623: Fri Jun 21 19:07:15 2013



Lab 1 & 2 Summary

- **Today we covered:**
 - Easy spinning of a BLDC-Motor
 - Measuring BEMF-Signals
 - Measuring Motor Current for Controlling the Efficiency



Lab 3: Adjusting Controls during Runtime

Lab 3 Summary

- **Today we covered:**
 - Every Motor is possible
 - Efficiency can increase
 - No Software-Change by changing the Motor !!

Additional Resources

- **AN1160 Sensorless BLDC with dsPIC[®] DSCs**
- **AN1305 Sensorless BLDC with PIC16F163X**

Dev Tools For This Class

- **DV244005 MPLAB® REAL ICE™ In-circuit emulator**
- **DM330021-2 MCLV-2 Board**
- **MAxxxxxx PIM-Module PIC16F1783**
- **AC300020 BLDC-Motor, 24V**
- **AC002013 24V Power Supply**



LEGAL NOTICE

SOFTWARE:

You may use Microchip software exclusively with Microchip products. Further, use of Microchip software is subject to the copyright notices, disclaimers, and any license terms accompanying such software, whether set forth at the install of each program or posted in a header or text file.

Notwithstanding the above, certain components of software offered by Microchip and 3rd parties may be covered by “open source” software licenses – which include licenses that require that the distributor make the software available in source code format. To the extent required by such open source software licenses, the terms of such license will govern.

NOTICE & DISCLAIMER:

These materials and accompanying information (including, for example, any software, and references to 3rd party companies and 3rd party websites) are for informational purposes only and provided “AS IS.” Microchip assumes no responsibility for statements made by 3rd party companies, or materials or information that such 3rd parties may provide.

MICROCHIP DISCLAIMS ALL WARRANTIES, WHETHER EXPRESS, IMPLIED, OR STATUTORY, INCLUDING ANY IMPLIED WARRANTIES OF NONINFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT WILL MICROCHIP BE LIABLE FOR ANY DIRECT OR INDIRECT, SPECIAL, PUNITIVE, INCIDENTAL, OR CONSEQUENTIAL LOSS, DAMAGE, COST, OR EXPENSE OF ANY KIND RELATED TO THESE MATERIALS OR ACCOMPANYING INFORMATION PROVIDED TO YOU BY MICROCHIP OR OTHER THIRD PARTIES, EVEN IF MICROCHIP HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES OR THE DAMAGES ARE FORESEEABLE.

TRADEMARKS:

The Microchip name and logo, the Microchip logo, dsPIC, FlashFlex, KEELOQ, KEELOQ logo, MPLAB, PIC, PICmicro, PICSTART, PIC³² logo, rPIC, SST, SST Logo, SuperFlash and UNI/O are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

FilterLab, Hampshire, HI-TECH C, Linear Active Thermistor, MTP, SEEVAL and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Silicon Storage Technology is a registered trademark of Microchip Technology Inc. in other countries.

Analog-for-the-Digital Age, Application Maestro, BodyCom, chipKIT, chipKIT logo, CodeGuard, dsPICDEM, dsPICDEM.net, dsPICworks, dsSPEAK, ECAN, ECONOMONITOR, FanSense, HI-TIDE, In-Circuit Serial Programming, ICSP, Mindi, MiWi, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, mTouch, Omniscent Code Generation, PICC, PICC-18, PICDEM, PICDEM.net, PICKit, PICtail, REAL ICE, rFLAB, Select Mode, SQL, Serial Quad I/O, Total Endurance, TSHARC, UniWinDriver, WiperLock, ZENA and Z-Scale are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

GestIC and ULPP are registered trademarks of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

All other trademarks mentioned herein are property of their respective companies.